Homework 4 Report

Tiantu Xu

Part A

API (`MyImg2Num`)

```
[nil] train()
[int] forward([28x28 ByteTensor] img)
```

How to run:

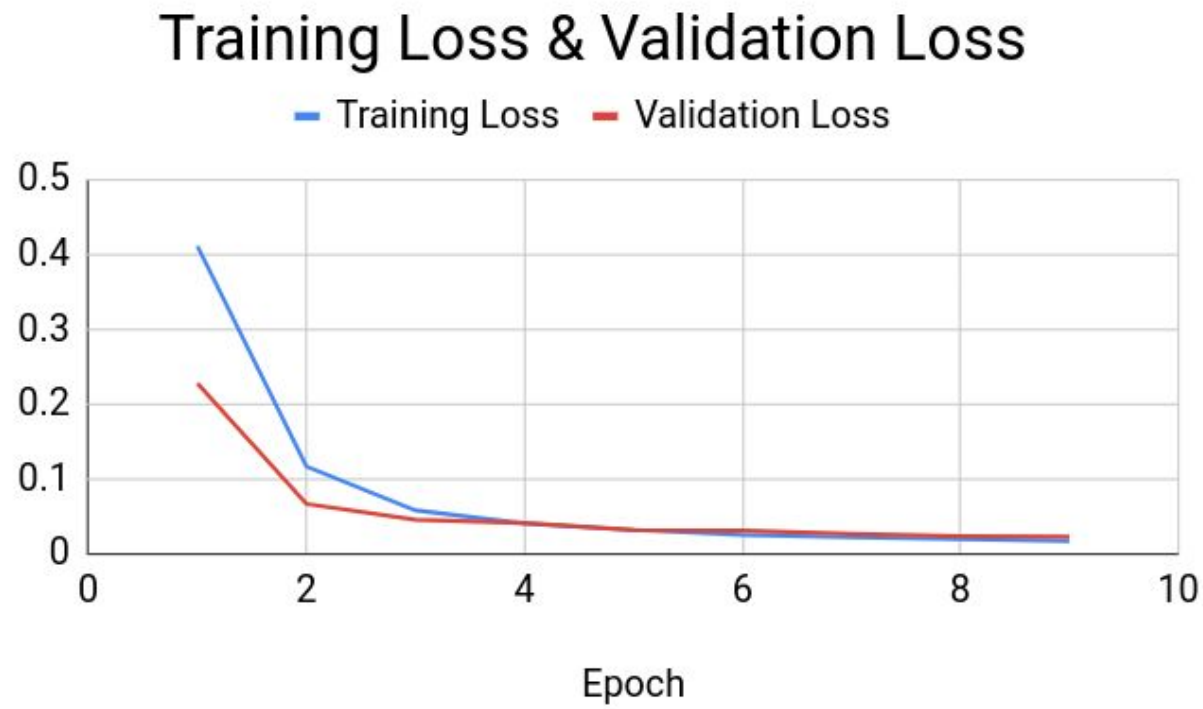
$ python my_img2num.py

Results:

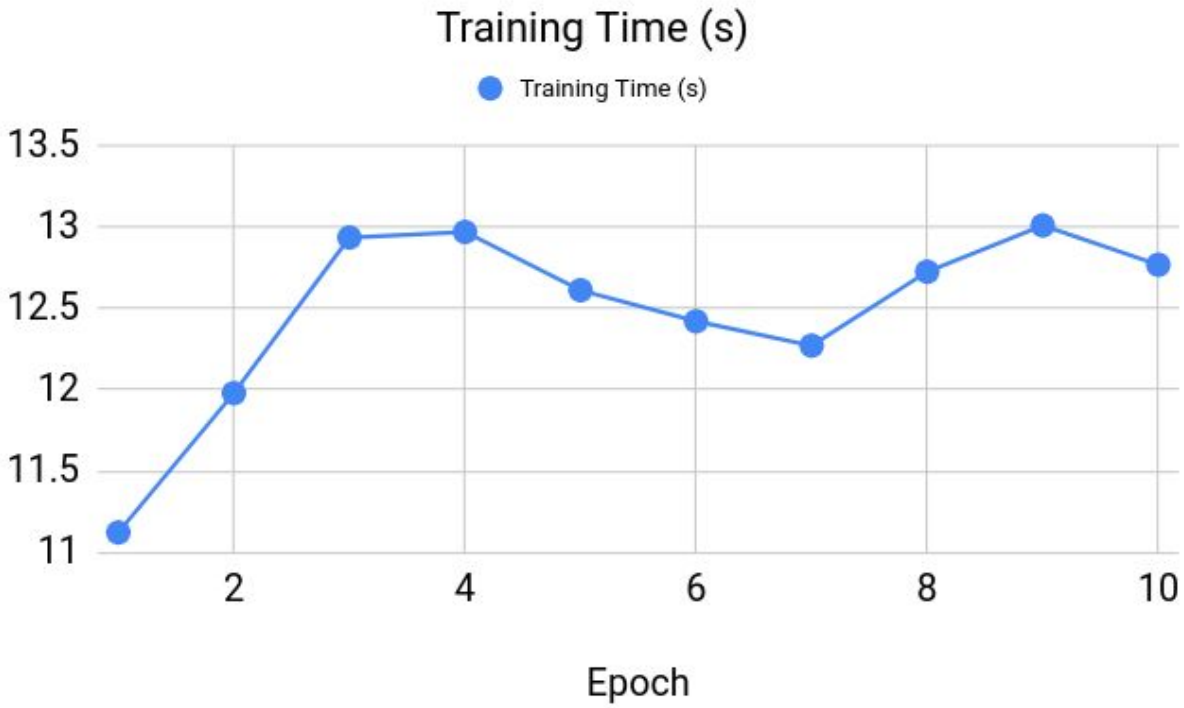
MyImg2Num (10 epochs, learning_rate = 0.1, batch = 10)

Epoch	Training Loss	Validation Loss	Training Time (s)	Accuracy
1	0.410908528646	0.227412890625	11.125524044	0.679
2	0.116753896077	0.0667981994629	11.9799752235	0.921
3	0.0582067159017	0.0455808319092	12.9339601994	0.949
4	0.0408795649211	0.041536416626	12.9679269791	0.941
5	0.0321672515869	0.0316858215332	12.6108791828	0.957
6	0.0254776713053	0.0312508148193	12.4203910828	0.96
7	0.0221015650431	0.0266047210693	12.2711939812	0.96
8	0.0195165328979	0.0235853439331	12.7247190475	0.965
9	0.0170987256368	0.0232279052734	13.0080499649	0.973
10	0.0152412541707	0.0216670059204	12.7670741081	0.965

Training loss vs validation loss



Training Time



Part B:

API (`NnImg2Num`)

```
[nil] train()
[int] forward([28x28 ByteTensor] img)
```

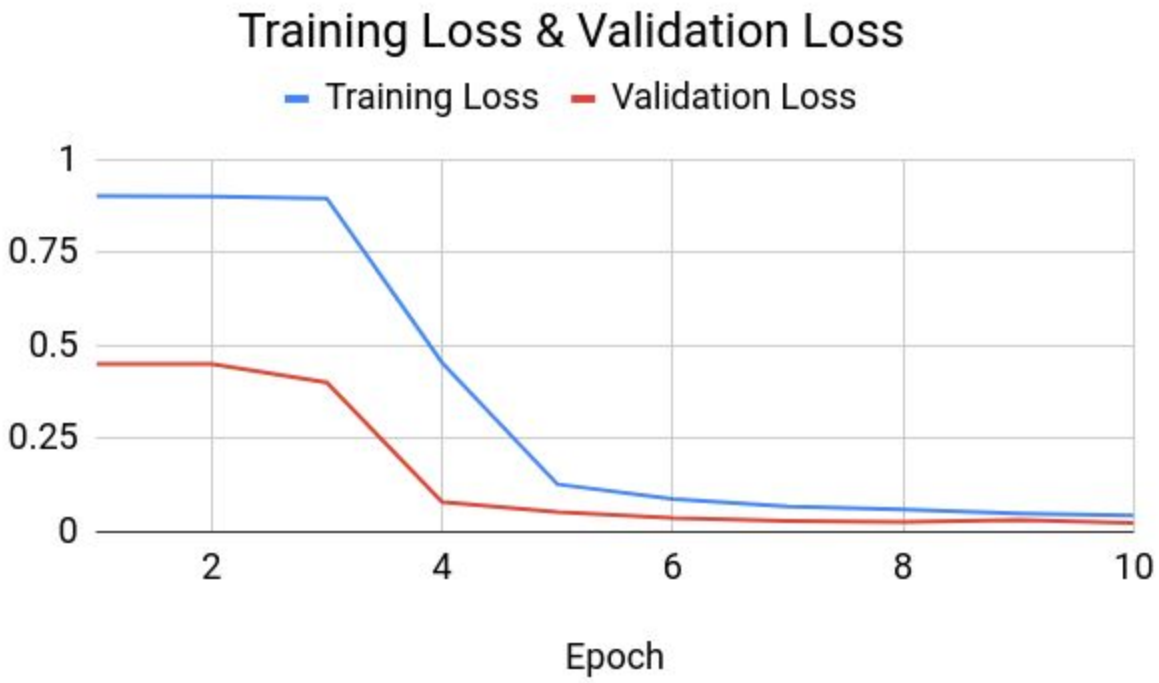
How to Run:

$ python nn_img2num.py

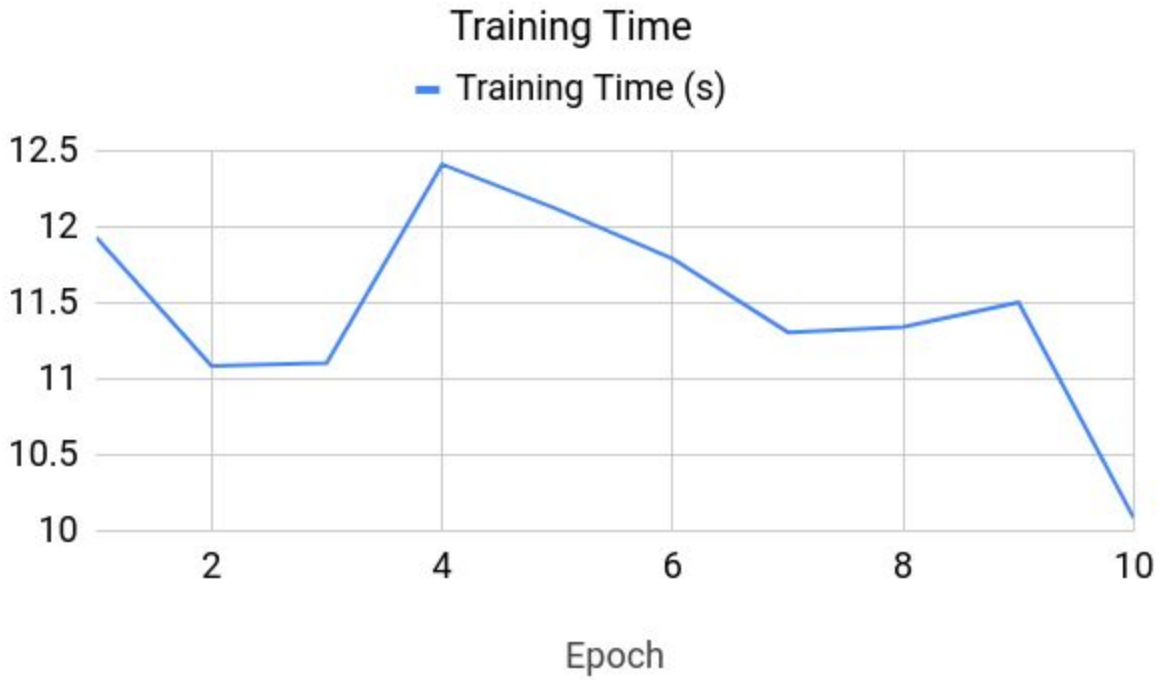
NnImg2Num (10 epochs, learning_rate = 15, batchBoth my_ing2num and nn_img2num has Both my_ing2num and nn_img2num has = 10)

Epoch	Training Loss	Validation Loss	Training Time (s)	Accuracy
1	0.9022158692	0.450721728516	11.9309060574	0.113
2	0.9009830082	0.450598291016	11.0853738785	0.094
3	0.895418866	0.401083447266	11.1050429344	0.291
4	0.4541737743	0.080305279541	12.4098210335	0.9
5	0.127837248	0.053724029541	12.1142401695	0.939
6	0.08875948812	0.0367151397705	11.7884788513	0.966
7	0.0678859256	0.0287972259521	11.3057489395	0.96
8	0.06006281138	0.0266293518066	11.3400840759	0.967
9	0.04908724283	0.0325099884033	11.5019719601	0.952
10	0.0446912301	0.0239268936157	10.0839180946	0.972

Training loss vs validation loss



Training Time



Conclusion:

Both my_ing2num and nn_img2num has descending training error and validation error. Training time is different every time.